

Attorney Docket No. 42P17993

APPLICATION FOR UNITED STATES PATENT

FOR

PASS THROUGH DEBUG PORT ON A HIGH SPEED ASYNCHRONOUS LINK

Inventor: Ashish Gupta
Bahaa Fahim
Kent Dickey
Jonathan Jasper

Prepared by: Calvin E. Wells
Reg. No. 43,256

PASS THROUGH DEBUG PORT ON A HIGH SPEED ASYNCHRONOUS LINK

Field of the Invention

[0001] The present invention pertains to the field of computer systems. More particularly, this invention pertains to the field of communicating debug information between components within a computer system.

Background of the Invention

[0002] The ability to debug computer system components is an important capability that allows computer system component engineers and technicians to identify problems and to improve their products. Many computer system components include circuitry to generate debug information that can be delivered to a debug port that may include several pins on the components. The debug information can be accessed by observing the activity on the pins with a logic analyzer. A disadvantage of this approach is that several, perhaps as many as 16 or more, pins are needed for the debug port. The addition of these pins for a debug port results in increased die and package size, as well as component cost and lower silicon yield.

Brief Description of the Drawings

[0003] The invention will be understood more fully from the detailed description given below and from the accompanying drawings of embodiments of the invention which, however, should not be taken to limit the invention to the specific embodiments described, but are for explanation and understanding only.

[0004] Figure 1 is a block diagram of one embodiment of a computer system including a north bridge having a memory interface controller coupled to an extended memory bridge.

[0005] Figure 2 is a block diagram of one embodiment of a portion of a memory interface controller.

[0006] Figure 3 is a block diagram of one embodiment of a portion of an extended memory bridge.

Detailed Description

[0007] In general, one embodiment of a computer system includes a first bridge device that includes an interface controller. The interface controller combines debug information generated within the bridge device with a training pattern. The first bridge device is coupled to a second bridge device via a high-speed asynchronous interconnect. The first bridge device converts the debug information and training pattern into a packet to be transmitted over the interconnect to the second bridge device. The training pattern serves to allow the second bridge device to maintain bit and symbol synchronization during the transfer of the debug information. The second bridge device receives the packet of combined debug information and training pattern and separates the debug information from the training pattern. The debug information may then be output to a memory bus where the debug information can be observed by a logic analyzer.

[0008] Figure 1 is a block diagram of one example embodiment of a computer system 100 including a north bridge 110 having a memory interface controller 200 coupled to an extended memory bridge (XMB) 300 via a memory interface bus 115. The memory interface 115 is a high-speed asynchronous link. For this example embodiment, the memory interface 115 is 10 bits wide and operates at a clock speed in the range of 2.1-3.2 GHz. The computer system 100 includes processors 102, 104, 106, and 108 coupled to the north bridge 110. The north bridge is further coupled to XMBs 120, 130, and 140. Each of the XMBs may be coupled to memory devices. The XMB 300 is coupled to a double data rate (DDR) memory bus 125.

[0009] The example computer system 100 is only one of a wide variety of possible computer system configurations. Further, although the embodiments described herein discuss a DDR memory bus, other embodiments are possible with other memory types.

[0010] The north bridge 110 includes circuitry (not shown) for generating debug information. Any of a wide variety of techniques and methods for generating or gathering debug information are possible in this embodiment. The interface controller 200 combines the generated debug information and a training pattern into a packet for transmission over the DDR bus 125 to the XMB 300. Because the memory interface 115 is asynchronous (the clock information is derived from edge transitions in the transmitted data), a training pattern is transmitted with the debug information in order to ensure that enough data transitions occur on the 10 wires of the interface 115 to allow the XMB 300 to maintain bit and symbol synchronization during the transfer of the debug information.

[0011] Before the memory interface controller 200 can deliver the debug information to the XMB 300, it first transmits a series of control packets that alert the XMB to the impending transfer of debug information. For this example embodiment, the debug information transfer is accomplished outside of the normal memory interface protocol. The XMB 300 therefore needs to be informed as to when the debug information transfer is to take place so that the XMB 300 can treat the received information appropriately.

[0012] The control packets that are transmitted from the memory interface controller 200 to the XMB 300 may be formatted like that shown in Table 1, below.

W9	W8	W7	W6	W5	W4	W3	W2	W1	W0
----	----	----	----	----	----	----	----	----	----

CRC	CRC	CRC	Control Packet	10
CRC	CRC	CRC	Control Packet	9
CRC	CRC	CRC	Control Packet	8
CRC	CRC	CRC	Control Packet	7
CRC	CRC	CRC	Control Packet	6
CRC	CRC	CRC	Control Packet	5
CRC	CRC	CRC	Control Packet	4
CRC	CRC	CRC	Control Packet	3
CRC	CRC	CRC	Control Packet	2
CRC	CRC	CRC	Control Packet	1

Table 1 – Control Packet Formatting

[0013] Table 1 shows 10 control packets that are to be sent consecutively from the memory interface controller 200 to the XMB 300. Bit 0 (W0, with “W” representing “wire”) of each of the packets includes an offset number that tells the XMB how many control packets will be issued before transmission of the debug information packets. For example, if the value at the W0 position is a 1, then the XMB should expect that the next packet will be debug information. The offset value begins at 10 and counts down with each successive control packet. The multiple control packets are sent in order to ensure that at least one of the packets will be transmitted and received without error. There is no reply mechanism for the XMB 300 to indicate a successful transmission, so the multiple packets provide redundancy to ensure at least one control packet is received without error.

Each of the control packets includes several cyclic redundancy check (CRC) bits to allow the XMB 300 to determine whether the packets are successfully received.

[0014] For this example embodiment, the debug and training pattern information is organized into 80 bit packets, where 72 bits are debug information and 8 bits are the training pattern. Table 2 below shows one possible way to organize the debug and training information packets (10 packets are shown).

W9	W8	W7	W6	W5	W4	W3	W2	W1	W0
Train	71:64	62:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
71:64	Train	62:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
63:56	71:64	Train	55:48	47:40	39:32	31:24	23:16	15:8	7:0
55:48	71:64	62:56	Train	47:40	39:32	31:24	23:16	15:8	7:0
47:40	71:64	62:56	55:48	Train	39:32	31:24	23:16	15:8	7:0
39:32	71:64	62:56	55:48	47:40	Train	31:24	23:16	15:8	7:0
31:24	71:64	62:56	55:48	47:40	39:32	Train	23:16	15:8	7:0
23:16	71:64	62:56	55:48	47:40	39:32	31:24	Train	15:8	7:0
15:8	71:64	62:56	55:48	47:40	39:32	31:24	23:16	Train	7:0
7:0	71:64	62:56	55:48	47:40	39:32	31:24	23:16	15:8	Train

Table 2 – Debug and Training Information Packet Organization

[0015] The 80 bit packets are sent 10 bits at a time for 8 consecutive clock periods. 10 packets are sent consecutively. The 8 bits of training information are moved to a

different wire for each packet, thereby ensuring that each wire receives a training pattern during the string of 10 packets.

[0016] The training packets may also have an advantage of providing parity information for each of the packets. For example, an 8-bit training pattern of 1010_1010 can be transmitted if the parity for the packet is even, or a pattern of 0101_0101 can be transmitted if the parity for the packet is odd.

[0017] The XMB 300 receives the packet of combined debug information and training pattern and separates the debug information from the training pattern. The debug information may then be output to a memory bus 125 where the debug information can be observed by a logic analyzer 150.

[0018] Figure 2 is a block diagram of one embodiment of a portion of the memory interface controller 200. A debug information assembly unit 240 receives a training pattern 201 and debug data 205. The assembly unit 240 also receives input from a train counter 210. The assembly unit 240 generates an 80-bit packet including 72 bits of debug information and 8 bits of training pattern. The training pattern is placed in an appropriate location according to position indicated by the train counter 210. The debug and training information packet is received by a multiplexer 250 which also receives normal memory interface traffic 203 and control packets from a control packet sequence unit 230. The output of the multiplexer 250 is delivered to a serializer unit 220. The serializer 220 takes the 80 bit packets and reduces them down to 10 wires for transfer across the memory interface 115.

[0019] Figure 3 is a block diagram of one embodiment of a portion of the extended memory bridge 300. A debug and training pattern packet is received over the memory

interface 115 at a de-serializer unit 310. The de-serializer unit 310 delivers the received packet to a debug data extraction unit 320 that separates the debug data from the training data. A train counter 330 indicates to the debug data extraction unit the location of the training pattern for the current packet. The debug data is then delivered to a buffer 350. The debug buffer 350 allows for the case where the memory interface 115 operates at a different clock speed than the DDR bus 125. A DDR memory controller 340 drives the debug data located in the buffer 350 onto the DDR bus 125, where the data can either be stored in memory or viewed by a logic analyzer coupled to the DDR bus 125.

[0020] In the foregoing specification the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than in a restrictive sense.

[0021] Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments, of the invention. The various appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments.